

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

## Zadání bakalářské práce

Student: **Denisa Smékalová**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**  
**Individual Professional Practice in the Company**

Jazyk vypracování: čeština

### Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Tieto Czech s.r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

### Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Dohnálek**


Konzultant bakalářské práce: Ing. Michal Laš

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018

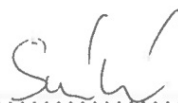
  
doc. Ing. Jan Platoš, Ph.D.  
vedoucí katedry



  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární  
prameny a publikace, ze kterých jsem čerpala.

V Ostravě 2018

.....  


Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a  
zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

**Tieto Czech s.r.o.**  
28. října 3346/91

702/00 Ostrava - Moravská Ostrava  
IČO 64608051 DIČ CZ64608051

V Ostravě 17. dubna 2018

.....

Ráda bych na tomto místě poděkovala všem, kteří mi s touto prací pomohli. Své rodině za podporu při studiu, společnosti Tieto za poskytnutí praxe a panu Ing. Michalovi Laši za veškerou pomoc v práci.

## **Abstrakt**

Tato bakalářská práce pojednává o absolvování individuální odborné praxe ve společnosti Tieto Czech s.r.o. Cílem bylo vytvořit nové reportovací řešení pro nástroj Weblist, a také vytváření reportů pro dashboard VSTS. Práce obsahuje popis odborného zaměření společnosti a mého pracovního zařazení. Následuje přiblížení použitých technologií. Další kapitoly se zabývají konkrétními postupy, které vedou k vyřešení úkolu.

**Klíčová slova:** praxe, Tieto, SQL, business intelligence

## **Abstract**

This bachelor thesis concerns with completion of individual professional practice in Tieto Czech i.n.c. The goal was to create a new reporting solution for the Weblist tool, as well as to create reports for the VSTS dashboard. This thesis contains a description of the company's professional orientation and my job description, followed by description of used technologies. Next chapters concern with methods used to solve the problem.

**Key Words:** practice, Tieto, SQL, business intelligence

# Obsah

|                                                   |           |
|---------------------------------------------------|-----------|
| Seznam použitých zkratk a symbolů                 | 8         |
| Seznam obrázků                                    | 9         |
| Seznam výpisů zdrojového kódu                     | 10        |
| <b>1 Úvod</b>                                     | <b>11</b> |
| <b>2 Společnost a popis pracovní pozice</b>       | <b>12</b> |
| 2.1 O společnosti . . . . .                       | 12        |
| 2.2 Pracovní pozice, náplň a prostředí . . . . .  | 12        |
| <b>3 Přehled použitých technologií a znalostí</b> | <b>13</b> |
| 3.1 Business intelligence . . . . .               | 13        |
| 3.2 Reporting . . . . .                           | 13        |
| 3.3 SQL . . . . .                                 | 13        |
| 3.4 SSRS . . . . .                                | 13        |
| 3.5 Power BI . . . . .                            | 14        |
| 3.6 OData . . . . .                               | 14        |
| 3.7 Weblist . . . . .                             | 14        |
| 3.8 VSTS . . . . .                                | 14        |
| <b>4 Zadané úkoly</b>                             | <b>16</b> |
| <b>5 Hlavní úkol a postup řešení</b>              | <b>17</b> |
| 5.1 Příprava . . . . .                            | 17        |
| 5.2 Databáze . . . . .                            | 17        |
| 5.3 Implementace zadání . . . . .                 | 18        |
| 5.4 Přejít na Power BI . . . . .                  | 20        |
| 5.5 Trend Analysis . . . . .                      | 23        |
| 5.6 Porovnání starého a nového řešení . . . . .   | 26        |
| 5.7 Budoucí funkce . . . . .                      | 27        |
| <b>6 Další úkoly</b>                              | <b>28</b> |
| 6.1 KPI Widgety . . . . .                         | 28        |
| 6.2 VSTS reporty . . . . .                        | 30        |
| <b>7 Závěr</b>                                    | <b>32</b> |
| <b>Literatura</b>                                 | <b>33</b> |

## Seznam použitých zkratek a symbolů

|       |                                   |
|-------|-----------------------------------|
| TIPS  | – Tieto Integrated Paper Solution |
| BI    | – business intelligence           |
| SQL   | – Structured Query Language       |
| SSRS  | – SQL Server Reporting Services   |
| DAX   | – Data Analysis Expressions       |
| OData | – Open Data Protocol              |
| REST  | – Representational State Transfer |
| HTTP  | – The Hypertext Transfer Protocol |
| JSON  | – JavaScript Object Notation      |
| ID    | – identity                        |
| KPI   | – Key Performance Indicator       |



## Seznam obrázků

|   |                                                                                 |    |
|---|---------------------------------------------------------------------------------|----|
| 1 | Zjednodušený relační model . . . . .                                            | 18 |
| 2 | Ukázka SSRS reportu s parametry . . . . .                                       | 20 |
| 3 | Ukázka Power BI reportu . . . . .                                               | 21 |
| 4 | Power BI Report Server s reporty . . . . .                                      | 22 |
| 5 | Původní verze reportovacího systému . . . . .                                   | 26 |
| 6 | Porovnání tabulek při načítání pomocí konektoru OData (nalevo) a VSTS (napravo) | 29 |
| 7 | Kombinovaný graf pro VSTS . . . . .                                             | 31 |
| 8 | Report porovnání hodin pro VSTS . . . . .                                       | 31 |

## Seznam výpisů zdrojového kódu

|   |                                                                    |    |
|---|--------------------------------------------------------------------|----|
| 1 | Ukázka jednoduchého dotazu . . . . .                               | 23 |
| 2 | Výsledný dotaz pro výběr vytvořených a uzavřených tiketů . . . . . | 24 |
| 3 | Kód pro vytvoření tabulky . . . . .                                | 24 |
| 4 | Dotaz pro výběr otevřených tiketů . . . . .                        | 25 |
| 5 | Přiřazení hodnot k datové tabulce . . . . .                        | 25 |
| 6 | Kód pro součet effort bodů . . . . .                               | 30 |

# 1 Úvod

V dnešní době požaduje čím dál více firem před nástupem na pozici předchozí zkušenosti. Školy se ale hlavně zaměřují na teoretické znalosti a praktická část výuky se vůbec nepodobá skutečným postupům, které se využívají ve firmách. Požadovanou praxi tak není zcela jednoduché získat.

Proto se mi velmi zamlouvala možnost zvolit si vypracování bakalářské práce formou odborné praxe u firmy. Z mého pohledu se jednalo o jedinečnou příležitost, jak získat cenné zkušenosti potřebné pro mé budoucí povolání, a také náhled toho, jak taková práce v IT firmě skutečně vypadá. Mimo jiné jsem si chtěla vyzkoušet, jak obstojí mé dosavadní znalosti v praxi.

Hlavním cílem této bakalářské práce je seznámit čtenáře s průběhem mé odborné stáže u firmy Tieto, kde jsem pracovala na pozici junior developer. Během stáže bylo mým cílem vytvořit nové reportovací řešení, které má nahradit stávající zastaralý způsob předávání informací, a to za pomoci moderních technologií, které se zabývají touto problematikou. Tento úkol mi byl udělen na míru, jelikož jsem u pohovoru projevila zájem o analýzu a zpracování dat.

Následující kapitoly mají za úkol kromě detailního popisu mé praxe čtenáři přiblížit mé pracovní zařazení a seznámit s technologiemi, jež byly nedílnou součástí při vykonávání mé práce. Jedna kapitola je také věnována vedlejším úkolům, které jsem plnila v době, kdy jsem čekala na další zadání či zpětnou vazbu k mému reportovacímu systému.

## 2 Společnost a popis pracovní pozice

### 2.1 O společnosti

Společnost Tieto je největší severoevropská IT společnost, působící na českém trhu od roku 2001. Poskytuje komplexní IT servis, dále také zajišťuje služby v oblasti vývoje produktů pro firmy působící v odvětví komunikace a integrovaných technologií. Díky svým znalostem, technologickým vizím a inovativnímu myšlení se společnost aktivně snaží inspirovat a zapojit své zákazníky do hledání nových způsobů pro zefektivnění jejich podnikatelské činnosti.

Tieto má své sídlo ve finských Helsinkách a zaměstnává přes 13 000 expertů z téměř 20 zemí světa. V roce 2004 bylo otevřeno softwarové centrum v Ostravě. S více než 2 200 zaměstnanci společnost patří mezi největší zaměstnavatele v oblasti IT služeb v České republice a největším v Moravskoslezském kraji. Dle počtu kmenových zaměstnanců je česká pobočka třetí největší pobočkou Tieto korporace na světě. První dvě místa zaujímají mateřské země Finsko a Švédsko.[1]

### 2.2 Pracovní pozice, náplň a prostředí

V rámci praxe jsem zastávala pozici junior developer. Jelikož jsem na pohovoru prohlásila, že by mě mohla bavit analýza, dostala jsem pozici zaměřenou na business intelligence.

Byla jsem přidělena do týmu TIPS, který se zabývá komplexním řešením pro papírenský průmysl. Mým úkolem bylo zjednodušit práci manažerů, a to vytvořením nového reportovacího systému pro nástroj Weblist. Mou další prací bylo vytváření podrobnějších widgetů pro dashboard VSTS. Tyto úkoly jsem zpracovávala sama, u některých komplexnějších problémů mi pomáhal můj konzultant Ing. Michal Laš.

Při nástupu jsem absolvovala školení nazvané Induction days, které nováčky seznamuje s firemní infrastrukturou, bezpečnostními zásadami a benefity, které mohou zaměstnanci využít.

Veškerou práci jsem prováděla na služebním notebooku, který mi byl přidělen v den nástupu. Potřebné programy jsem ale využívala na testovacím serveru, ke kterému jsem se připojovala pomocí vzdálené plochy. Komunikace s kolegy probíhala především osobně, někdy bylo třeba využít firemní Skype či e-mail, jelikož jsem potřebovala komunikovat s kolegy z německé pobočky Tieto.

### 3 Přehled použitých technologií a znalostí

Tato kapitola obsahuje popis veškerých použitých technologií a detailnější přiblížení některých pojmů vyskytujících se v této práci.

#### 3.1 Business intelligence

Business Intelligence (BI) je výraz pro procesy, aplikace, znalosti, platformy, technologie a nástroje, které podporují porozumění datům, jejich vztahům a trendům. Pro BI neexistuje jednotná definice, ale obecně lze říci, že představuje specifický typ úloh podnikové informatiky. Tyto úlohy podporují analytické, plánovací a rozhodovací činnosti.

Nástroje pro BI využívají matematického aparátu a grafické interpretace tak, aby na základě těchto výstupů získali manažeři rychlejší a lepší přehled o chodu společnosti, nebo průběhu projektu apod. a mohli tak činit správná rozhodnutí i na základě velkého množství dat.[2][3]

#### 3.2 Reporting

Reporting jsou činnosti spojené s dotazováním se do databází pomocí jejich standardních rozhraní. Cílem je poskytovat včas a ve vhodné formě podklady (statické nebo dynamické) pro podporu rozhodování na všech stupních organizační struktury.

Podmnožinou reportingu jsou tzv. dashboardy, které jsou dynamické – interaktivní budíky, grafy, tabulky zaměřené na vzhled a koncentrovanou podobu na jednu obrazovku.[3]

#### 3.3 SQL

Structured Query Language (SQL) je strukturovaný dotazovací jazyk. Je to nástroj používán k manipulaci, správě a organizování dat uložených v databázi. Jedná se o neprocedurální jazyk – vyjadřuje, co chceme provést.

Kromě dotazování slouží SQL také k definování dat, např. určení struktury tabulky a její následovné naplnění záznamy nebo také můžeme definovat vztahy mezi tabulkami apod.[4][5]

Ve své práci jsem s jazykem SQL pracovala jen za účelem dotazování, následně získaná data byly zpracovány dalšími nástroji popsány níže.

#### 3.4 SSRS

SQL Server reporting services (SSRS) je serverově orientovaný systém pro tvorbu reportů od firmy Microsoft. Je součástí Microsoft SQL server services.

Systém je využíván pro přípravu nejrůznějších interaktivních či tištěných reportů. Pro práci je poskytováno rozhraní v Microsoft Visual Studio, díky čemuž se mohou vývojáři a také SQL administrátoři připojit k databázím a využívat tak poskytovaných služeb ke zpracování reportů různými způsoby.[6]

### 3.5 Power BI

Power BI je poměrně nový nástroj, který je taktéž vytvářen pod záštitou firmy Microsoft. Slouží pro analýzu a prezentaci dat. Díky obsaženým reportingovým a analytickým nástrojům lze zpracovat i větší objem dat. Získané informace lze poté jednoduchým a interaktivním způsobem prezentovat ve formě vizuálních prvků na jakémkoli zařízení.[7]

Nástroj stále není zcela dokončen, a proto nové funkce přibývají každý měsíc, z nichž některé dokázaly vyřešit problémy v mé práci nebo přinesly užitečný doplněk.

#### 3.5.1 DAX

Data Analysis Expressions (DAX) je knihovna funkcí, operátorů a konstant, které mohou být kombinovány pro sestavení vzorců či výrazů, které se používají k výpočtům a vrácení jedné nebo více hodnot v programech Microsoft SQL Server Analysis Services, Power Pivot v Excel, a Power BI Desktop.[8]

### 3.6 OData

Open Data Protocol (OData) je protokol pro dotazování a aktualizaci dat s využitím stávajících webových protokolů. Je založen na protokolech REST a na standardizovaných technologiích, např. HTTP a JSON. Poskytuje jednotný způsob, jak popisovat data i datový model. Protokol je považován za flexibilní technologii umožňující spolupráci mezi různými zdroji dat, aplikacemi, službami a klienty.[9]

### 3.7 Weblist

Weblist je program vyvíjený ve firmě Tieto. Jedná se o reportovací nástroj používaný v rámci projektů TIPS a SAP. Využívá se na zpětnou vazbu od zákazníků. Ti mohou skrze Weblist nahlásit jakýkoli problém či zaslat dotaz nebo návrh související s jejich TIPS/SAP produktem nebo službou.

Dále se nástroj využívá pro sledování tiketových řešení, testování a instalační procesy, a také pro jakoukoli komunikaci mezi zákazníkem a support týmem.

Weblist se používá také jako tiketový informační zdroj pro zákazníky v případě, že se problém nebo požadavek vyskytuje opakovaně, což šetří čas při řešení tiketů.

### 3.8 VSTS

Visual studio team services (VSTS) je cloudový servis pro vývojáře, sloužící ke spolupráci na kódu.

VSTS umožňuje rychlé a snadné plánování, vytváření a dodávání softwaru napříč platformami. Obsahuje integrované funkce, ke kterým lze přistupovat prostřednictvím webového prohlížeče. Další funkce lze přidat skrze doplňky. Užitečnou součástí je například dashboard, což je

nástěnka, na kterou mohou být umísťovány widgety<sup>1</sup> pro sdílení informací, zobrazení postupu práce apod.[10]

V Tietu VSTS nahradilo starší servis JIRA. Používá se pro spolupráci na projektech a pro zefektivnění práce, jelikož je vývoj postaven na metodě agilního vývoje softwaru.

---

<sup>1</sup>vizuální prvek, např. ve formě karty, grafu nebo obrázku

## 4 Zadané úkoly

V rámci mé praxe jsem dostala samostatný hlavní úkol. Cílem bylo vytvořit reportovací řešení, které bude využito na nástroji Weblist. Zpočátku bylo založeno na technologii SSRS, později jsem dle požadavků přešla na novější a efektivnější Power BI. Postup mého řešení je rozepsán v následující kapitole.

Nové reportovací řešení má hlavně pomoci manažerům Tieta k lepší distribuci dat. Pro své pracovní potřeby a také pro zákazníky potřebují zobrazit pohyb tiketů, z nichž vyplývají informace o kvalitě poskytovaného produktu. Aktuální řešení tyto informace předává, přístup k datům je však pomalý a neefektivní, také obsahuje technické problémy, což někdy může způsobit nefunkčnost. Z těchto důvodů je třeba vytvořit nové řešení, které odstraní nedostatky, poskytne rychlejší informace o stavech řešených tiketů a délky jejich trvání, nabídne efektivnější stromové procházení na sobě závislých tiketů a také moderní minimalistické interaktivní prostředí pro okamžité zjištění potřebné informace.

Kromě tohoto hlavního úkolu jsem také dostávala další menší úkoly, které se týkaly zpracování widgetů pro VSTS. Tyto widgety se dají zpracovat přímo pomocí nástrojů ve VSTS, některé detailnější ale potřebují zpracovat jinou cestou. Právě tyto složitější verze widgetů jsem dostala na starost.



## 5 Hlavní úkol a postup řešení

V této kapitole popisuji, jakým způsobem jsem vytvářela reportovací řešení pro nástroj Weblistu. První část je věnována tvorbě reportů v programu SSRS a popisu jednotlivých reportů, druhá se zabývá tvorbou reportů v programu Power BI a změnám oproti předchozímu řešení.

### 5.1 Příprava

Prvním podstatným úkolem bylo naučit se pracovat v programu SSRS a bližší pochopení business intelligence. K těmto účelům jsem využívala výuková videa na portálu Pluralsight. V rámci těchto tutoriálů jsem se učila i za pomoci výukových materiálů a zkoušela si příklady na ukázkových databázích.

Jako první jsem sledovala kratší video zaměřené na základní funkce SSRS, kde se reporty vytvářely hlavně přes Report Wizard – zjednodušené rozhraní, kde se pouze nastaví, co a jak chce uživatel zobrazit a program poté sám automaticky zpracuje např. tabulku nebo graf. Nejdůležitější z tohoto kurzu pro mě bylo pochopit a naučit se správně nastavit data source – zdroj dat, a dataset – dotaz nebo odkaz na entity, které vracejí hodnoty dat.[11] Data pro potřeby dataset se nevybírají úplně stejným způsobem jako pro běžný SQL dotaz. Většinou je třeba do dotazu přidat další hodnotu pro správné a přehledné zobrazení dat.

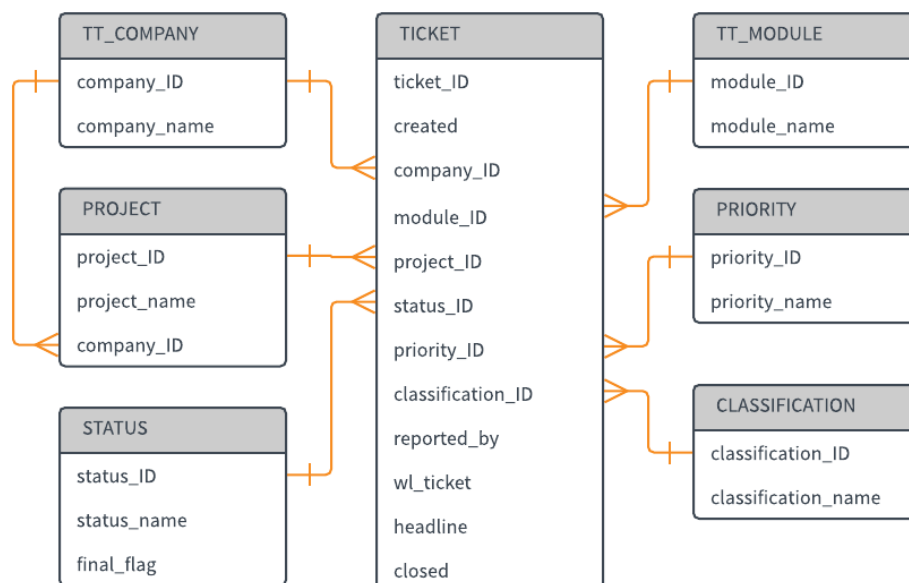
Další videokurz už byl mnohem obsáhlejší a věnoval se podrobnějšímu zpracování dat bez použití Report Wizard, což dává vývojáři mnohem větší svobodu v tom, jak chce svá data zobrazit. V tomto kurzu jsem se naučila, jak zacházet se všemi dostupnými nástroji v SSRS toolbox.

Kurz se zpočátku věnoval ručnímu vytváření tabulkových a grafových reportů. Další kapitoly byly věnovány jedné z nejdůležitějších částí programu – expressions. Pomocí expressions se dá nastavit, jak přesně se mají dané informace zobrazit či provádět různé výpočty apod. Poslední kapitoly tohoto kurzu se věnovaly vytváření a používání parametrů.

### 5.2 Databáze

Dalším krokem bylo seznámit se s databází Weblistu. Ze začátku jsem pracovala pouze s databází určenou pro vývoj. Ta je strukturálně stejná jako produkční, liší se pouze v obsažených datech.

K databázi neexistuje dokumentace a celkově má asi přes 200 tabulek. Jelikož bylo potřeba se zabývat pouze její malou částí, která se týká tiketů, zaslal mi vedoucí pro zorientování dotaz, jehož výsledek zobrazoval hlavní údaje k tiketům. Díky tomuto dotazu bylo snadné zjistit, které tabulky jsou pro mě důležité a dokázala jsem si tak vytvořit vlastní zjednodušený relační model.



Obrázek 1: Zjednodušený relační model

### 5.3 Implementace zadání

Zadání pro reporty jsem obdržela v podobě wordovského dokumentu, který obsahoval ukázky základních grafů a krátký popis toho, co mají přesně zobrazovat. Tyto grafy tvořily základ mých reportů. Každý z těchto grafů zobrazoval součet tiketů za kalendářní měsíc, třízených podle různých kritérií. Samozřejmě ne vždy byly informace v zadání dostačující a ohledně některých jsem se musela dále informovat. U veškerých uvedených parametrů jsem si sama zjišťovala, kde se nalézají v databázi.

U všech reportů jsem se držela stejné šablony. Na každém byla zobrazena tabulka a graf. Podstatnou částí byl graf, tabulka byla přidána hlavně pro kontrolu dat, nebyla součástí zadání. U každého reportu byl navíc použit různý typ tabulek, např. typ drilldown. Pro třízení dat podle měsíce jsem využívala sloupec created, z tabulky Ticket, který vyjadřuje přesný čas, kdy byl tiket vytvořen. Pro účely seřazení jsem sloupec v dataset nastavila přes funkci `to_char` tak, aby zobrazoval pouze měsíc a rok.

Jako první jsem se rozhodla zpracovat statistiku vytvořených položek, jelikož mi ze všech reportů v zadání přišel nejjednodušší a nejrychleji jsem objevila potřebná data pro sestavení v databázi. Tento report zobrazuje reporty podle klasifikace. Ta tikety třídí podle žádosti o změnu, problém, servisní požadavek a incident. Pokud je tiket označen jako incident, znamená to, že byla položka vytvořena zákazníkem. Typ klasifikace je v dotazu vybrán na základě ID položky, jedna položka se ale vyskytuje v tabulce vícekrát, což má za následek, že jedna položka má přiřazeno více než jedno identifikační číslo.

Další report – statistika otevřených položek je v základě stejný, jako výše popsany report statistiky vytvořených položek. Tikety zobrazuje také podle klasifikace, ale pouze ty, jejichž status má nastaveno *final flag* na 0, což znamená, že je tiket otevřený. Otevřené tikety jsou určeny stejným způsobem i u ostatních reportů.

Z těchto dvou vychází také report aktuálně otevřené tikety, který již podle názvu zobrazuje informace o všech tiketech, které jsou otevřené k aktuálnímu datu. Díky zobrazení podle klasifikace jde z koláčového grafu určit aktuální stav produktu. Například pokud by byl vysoký počet tiketů typu problém, znamenalo by to komplikace s jeho kvalitou.

Dále bylo třeba zpracovat report, který třídí otevřené tikety podle priority. Ta se dělí na střední, vysokou a kritickou. Pokud by se objevovalo hodně kritických incidentů, opět by to znamenalo problém s kvalitou produktu. Nejvíce incidentů vzniká při instalaci nových funkcí. Pokud osa grafu prudce klesá, znamená to úspěšné vyřešení problémů.

Poslední report zobrazuje počet tiketů podle modulů. Díky tomuto reportu mohou uživatelé vidět, v jaké oblasti jejich produktu se vyskytují problémy.

Po dokončení tohoto základního zobrazení bylo dalším úkolem přidat parametry pro vybrání firmy a projektu, který spadá pod danou společnost.

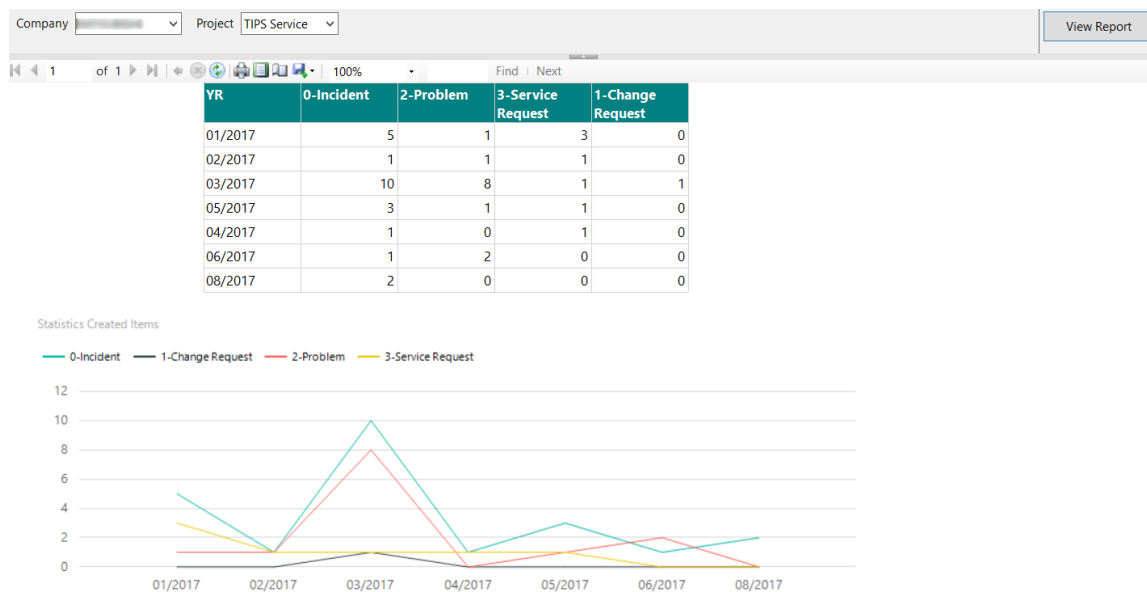
První bylo třeba ke každému datasetu přidat výběr firmy a projektu, což je pouze jednoduché rozšíření původního dotazu. Dalším krokem bylo vytvořit 2 nové datasety pro potřeby samotných parametrů, a to s výběrem ID a názvem projektů a firem. Jelikož byly všechny reporty vytvářeny v jednom souboru, mohl být pro zrychlení práce využit sdílený dataset – takovýto dataset se dá použít pro více reportů zároveň, není ho tedy třeba sestavovat pro každý report zvlášť.

Po těchto krocích už mohly být vytvořeny parametry. Nejprve je třeba nastavit hodnoty, které budou parametry zobrazovat. Ty mohou být zadány ručně nebo být načteny z datasetu. Jelikož jsem využila druhý případ, bylo třeba nastavit pole pro hodnotu(ID) a pro štítek, který uvidí uživatel.

Pro propojení parametrů s tabulkou už jen stačí v nastavení datasetu, který vybírá data pro report, přidat do podmínky **where** odkaz na parametr pomocí dvojtečky, a to následujícím způsobem: `Company_id = :company`. V záložce parametry u datasetu se už jen přiřadí správný parametr k odkazu z dotazu.

Parametry jsou automaticky umístěny na vrchní straně reportu. Výběr firmy a projektu se provádí pomocí rozbalovací nabídky.

Report není možné načíst bez zadání obou parametrů, pro jeho spuštění je nutné zmáčknout tlačítko pro zobrazení reportu v pravém horním rohu.



Obrázek 2: Ukázka SSRS reportu s parametry

## 5.4 Přechod na Power BI

Po přidání parametrů byly reporty připraveny k využití v praxi. V tu dobu se ale v týmu začal pro tvorbu reportů využívat nový program Power BI. Vedení poté rozhodlo, že zpracování v tomto programu by bylo pro reportovací řešení Weblistu vhodnější díky jeho interaktivním funkcím.

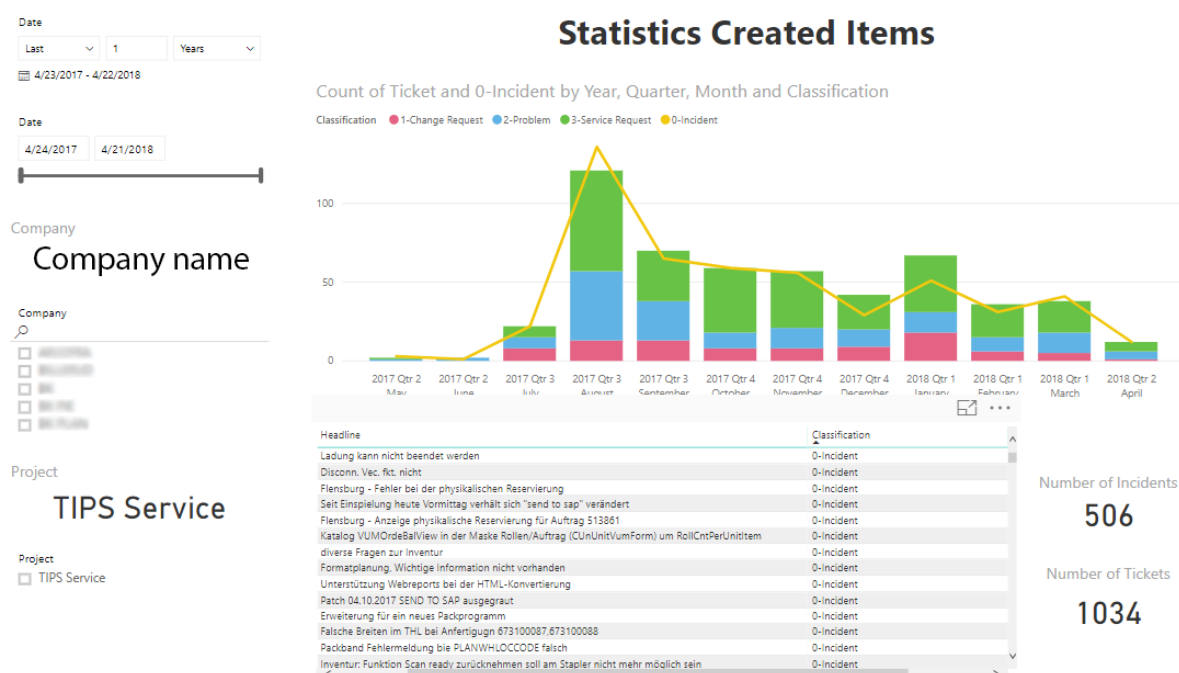
K pomocné orientaci v programu posloužila opět stránka Pluralsight, manipulace s Power BI je ale v porovnání s SSRS mnohem jednodušší a vyžaduje méně znalostí. Pro převedení reportů z SSRS stačilo minimum znalostí, vše ostatní jsem se učila až v průběhu práce s programem podle toho, co bylo třeba udělat.

Hlavní rozdíl mezi programy Power BI a SSRS je ve formě prezentace dat. SSRS je vhodnější na tištěnou formu reportů. Data jsou neměnná, jejich zobrazení ovlivňují pouze parametry. Power BI je vhodnější pro prezentaci dat, všechny prvky okamžitě reagují na změnu např. filtru.

Při vytváření reportů v Power BI také figuruje datasource a dataset. Práce s nimi je však lehce odlišná. Datasource se nastavují v nabídce aktuálně dostupných konektorů. Zdrojem dat nutně nemusí být databáze, dá se využít i soubor. Databáze nástroje Weblisu využívá Oracle, který Power BI podporuje. Při přechodu na tento program jsem začala využívat produkční databázi místo testovací, aby si mohly reporty vyzkoušet uživatelé a poskytnout mi tak zpětnou vazbu.

Nastavení dataset je spíše dobrovolné a není možné ho nastavit u všech konektorů. Například při připojení k Oracle databázi se vyskytuje ve formě, kdy při nastavování serveru se může pro výběr dat přidat SQL dotaz, jinak se databáze načte celá – potřebná data se poté vybírají pomocí různých nastavení a filtrů.

Pro jednoduchost jsem možnost nastavení dataset využila. Použila jsem stejné dotazy jako v případě SSRS se dvěma malými úpravami. Jelikož Power BI umí rozpoznat sloupec, který je typu date a ten pak následně převést na dny, měsíce apod., nebylo tak třeba využívat funkci `to_char` pro třídění podle měsíce. Další změna se týká výpočtu hodnot. Při vytváření grafu se dá jednoduše nastavit, jakou výpočetní operaci nad daty provést, nebylo tedy třeba využívat funkce `count` ani `group by`.



Obrázek 3: Ukázka Power BI reportu

Při designu jsem se opět u všech reportů snažila držet stejné šablony, tentokrát přizpůsobené funkcím Power BI. Na levé straně se nacházejí filtry. Ty slouží pro výběr data, firmy a projektu. Pro potřeby uživatelů se filtry pro výběr data vyskytují dva. Jeden je relativní – je výchozí a jeho účelem je, aby uživatel automaticky při otevření reportu zobrazoval tikety za poslední rok. Druhý datový filtr slouží pro zobrazení tiketů v jakémkoli časovém období – je třeba nastavit parametry od-do, nebo je možné využít posuvník.

Na filtru určeném pro zvolení firmy se vyskytuje vyhledávací pole, které bylo přidáno později na základě požadavku od uživatelů. Původně se firmy vybíraly z rolovacího seznamu. Tento způsob výběru dat ale nepodporuje vyhledávání, proto musel být změněn na seznam.

Na pravé straně se nachází graf a tabulka, která zobrazuje detailní informace o tiketech z grafu. Tabulka byla přidána na základě požadavků od uživatelů.

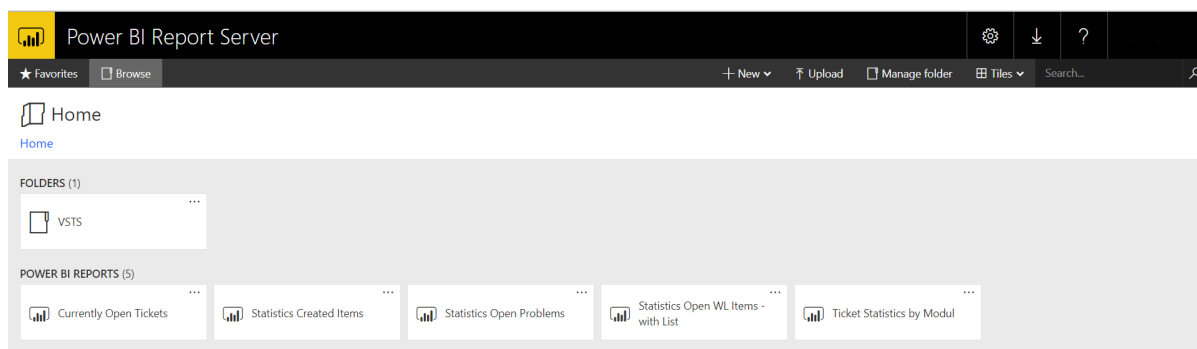
Části grafu mohou být označeny a tyto označené části se poté projevují na ostatních částech reportu, a to tak, že například tabulka zobrazí detailní informace o jednotlivých tiketech pouze ze zvýrazněné části, to samé platí o kartě, která zobrazuje počet tiketů. Report Statistika

vytvořených položek má navíc kartu, která zobrazuje počet incidentů. Takto uživatel může rychle zjistit, kolik přesně tiketů bylo vytvořeno zákazníky.

Graf v Power BI poskytuje různé interaktivní funkce. Například se dá nastavit drilldown na datum. Uživatel tak nemusí být nucen k zobrazení tiketů pouze podle měsíce, ale i podle roku či dne. Toto je jeden ze zásadních rozdílů oproti reportům v SSRS, kde byly data permanentně nastaveny na seskupení podle měsíce bez možnosti změny. Ta by šla pouze v případě změny nastavení dataset, konkrétně by bylo potřeba upravit funkci `to_char`. Dále je na grafu možnost data exportovat ve formátu csv, nebo zobrazit data v tabulce. Při této volbě se načte zobrazení, na kterém je pouze graf a tabulka. Tato tabulka ale ukazuje pouze všechny data z grafu. Označení určitých sloupců či os na grafu ji neovlivní.

Filtry se také nacházejí ve vysunovací liště na pravé straně reportu. Nevýhodou je, že pro lepší orientaci nových uživatelů je efektivnější umístění filtrů přímo na stránce reportu, kde si jich uživatel může všimnout hned po otevření reportu. Další nevýhodou těchto skrytých filtrů je, že na nich nelze nastavit vyhledávání, například tak jako na filtru pro výběr firmy. Tyto filtry jsou výhodné na reportu, kde je třeba filtrovat větší množství hodnot, jelikož se tak ušetří místo na stránce reportu. Dále také umožňují nastavit, zda filtrovaná hodnota ovlivňuje pouze určitý prvek reportu, celou stránku reportu, nebo pro celý soubor<sup>2</sup>.

Aby uživatelé měli k reportům pohodlný přístup, využila jsem k jejich publikaci Power BI Report Server, který jsem nainstalovala na testovací server. Publikace reportů probíhá jednoduše přes desktopovou verzi Power BI, kde reporty stačí jednoduše uložit na server místo do složky počítače. Server je přístupný z webového prohlížeče bez nutnosti zadávání jakýchkoli údajů, uživateli stačí pouze správná webová adresa.



Obrázek 4: Power BI Report Server s reporty

---

<sup>2</sup>Funkci lze využít, pokud soubor obsahuje několik stránek reportů.

## 5.5 Trend Analysis

Největším problémem bylo zobrazení trend analysis – trend analýzy. Trend analýza je jiné pojetí technické analýzy, které se snaží předvídat budoucí pohyby produktu na základě současných a minulých údajů. Je založena na myšlence, že to, co se stalo v minulosti dává podklad pro to, co se stane v budoucnu. Zahrnuje porovnání dat v průběhu po sobě jdoucího časového období za účelem zjištění vzoru nebo trendu.

Trend analýza nástroje Weblist zobrazuje otevřené (open), vytvořené (created) a zavřené (closed) tikety za měsíc během určité časové doby. Základem pro sestavení je výběr dat jako u každého jiného reportu – sestavení datasetu. Výběr kategorií jednotlivě pro výběr pomocí SQL je jednoduchý, problémem bylo dostat všechny data pouze z jednoho dotazu, aby se informace mohly zobrazit v jednom reportu.

Tvorbou tohoto reportu jsem se zabývala už v době, kdy jsem reporty vytvářela ještě v SSRS. Funkční řešení jsem našla až po přechodu na Power BI, jelikož mi to umožnilo rozdělit výběr dat na dva dotazy.

Jako první jsem řešila, jak dostat informace pro vytvořené a zavřené tikety v daném měsíci. Kategorie *created* je ta, jejíž status má nastaven *final flag* na 0, kategorie *closed* má naopak *final flag* 1. Problém tedy je, jak ve výsledku spočítat dvě hodnoty s různými parametry v jednom dotazu.

Pokud bych potřebovala vypočítat např. pouze kategorii *created*, stačilo by do podmínky *where* přidat vnořený select, který vybere všechny statusy, který mají přidělen *final flag* 0. Výsledný dotaz by poté vypadal takto:

---

```
SELECT
    count(t.ticket_id),
    t.created,
    f.company_name,
    p.project_name
FROM Ticket t
JOIN TT_company f ON t.company_id = f.company_id
JOIN Project p ON t.project_id = p.project_id
WHERE
    t.status_id in (select status_id from status where final_flag = 0)
GROUP BY t.created, f.company_name, p.project_name
```

---

Výpis 1: Ukázka jednoduchého dotazu

Pomocí běžného selectu nelze přidat tuto podmínku pro obě kategorie, hodnota by se vypočetla dohromady – v podmínce nelze říct, že tahle podmínka platí pro tento count a tato pro další count. Zkoušela jsem spoustu různých řešení jako například spojení dvou selectů, což nefungovalo. Nakonec jsem zjistila, že do *sum* při vybírání hodnot se dá přidat *case*. Řešení mi trvalo chvíli najít, jelikož jsem s *case* nikdy předtím nepracovala, ani o něm neslyšela. Výsledný dotaz byl potom krátký a velice jednoduchý. Do *case* stačilo vložit podmínku.

---

```

SELECT
    t.created,
    sum(case when s.final_flag = 0 then 1 else 0 end) as count_0,
    sum(case when s.final_flag = 1 then 1 else 0 end) as count_1,
    f.company_name,
    p.project_name
FROM ticket t
    JOIN status s ON s.status_id = t.status_id
    JOIN TT_company f ON t.company_id = f.company_id
    JOIN Project p ON t.project_id = p.project_id
GROUP BY t.created, f.company_name, p.project_name

```

---

Výpis 2: Výsledný dotaz pro výběr vytvořených a uzavřených tiketů

Další dotaz byl pouze pro spočítání otevřených tiketů. Tento problém byl ještě komplikovanější.

Každý tiket má dva časové údaje, datum, kdy byl vytvořen a kdy byl uzavřen. Do doby, než je uzavřen, se tiket považuje za otevřený. Na ose grafu se má tedy tiket spočítat v každém měsíci od otevření do uzavření, tzn. pokud byl tiket vytvořen 20.1. 2017 a uzavřen 12.4. 2017, musí se na ose objevit v lednu, únoru, březnu a dubnu. Tato osa je typický příklad trend analýzy.

Některé databáze (např. VSTS<sup>3</sup>) mají tabulku přímo pro potřeby zobrazení trendu tzv. snapshot. V těchto tabulkách se zobrazuje stav položky za každý den, lze tak tedy přesně určit, v jakém stavu se položka nacházela v určitou dobu. V nástroji Weblist se ale žádná takováto tabulka nevyskytuje a existují jen výše popsané údaje. Bylo tedy třeba bez existující snapshot tabulky zjistit, zdali tiket do daného měsíce patří či nikoli. Na tento problém jsem našla dvě různá řešení, kde výsledný report je v obou případech zhotoven díky možnostem v programu Power BI.

### 5.5.1 Řešení 1

První řešení vychází z možností Oracle databáze, kdy lze pomocí SQL vygenerovat datovou tabulku. Tabulka se vytváří mezi dvěma časovými obdobími – počáteční datum a poslední datum.

---

```

SELECT (to_date('01-JAN-2008','dd-mon-yyyy') + rownum -1) DayRecord
FROM all_objects
WHERE rownum <= to_date('01-JAN-2021','dd-mon-yyyy')-to_date('01-JAN-2008','dd-mon-yyyy')+1

```

---

Výpis 3: Kód pro vytvoření tabulky

K takto vzniklé datové tabulce bylo třeba správně přiřadit záznamy. Ty se přiřazují tak, že se porovnává záznam datové tabulky, který je mezi datem vytvoření tiketu a datem uzavření tiketu. Oracle databáze nemá funkci *between*, proto bylo třeba využít operátor větší/menší/rovno.

---

<sup>3</sup>Pro získání databáze je třeba mít nainstalovanou reportovací platformu Analytics Service



Jelikož některé tikety ještě nemusí být uzavřeny, nacházejí se tak v tabulce záznamy, které nemají vyplněnou kolonku *closed*. Aby se záznamy přiřadily správně, je sloupec *closed* ohraničen funkcí *nvl*, která hlídá, zda má záznam hodnotu *null*. Pokud ano, nahradí ho alternativní hodnotou, v tomto případě aktuálním datem.

---

```
SELECT Dates.DayRecord, COUNT(distinct t.ticket_id), f.company_name, p.project_name
FROM ticket t
JOIN
(
    SELECT (to_date('01-JAN-2008','dd-mon-yyyy') + rownum -1) DayRecord
    FROM all_objects
    WHERE rownum <= to_date('01-JAN-2021','dd-mon-yyyy')-to_date('01-JAN-2008','dd-mon-yyyy')+1
) Dates
on Dates.DayRecord >= t.created and Dates.DayRecord <= nvl(t.closed, systimestamp)
JOIN TT_company f ON t.company_id = f.company_id
JOIN Project p ON t.project_id = p.project_id
GROUP BY Dates.DayRecord, f.company_name, p.project_name
ORDER BY Dates.DayRecord desc
```

---

Výpis 4: Dotaz pro výběr otevřených tiketů

### 5.5.2 Řešení 2

Oba způsoby řešení se skládají ze stejných kroků, každé z nich je ale postaveno na jiné platformě. Druhé řešení vychází z možností PowerBI, které poskytuje vhodné nástroje k vytvoření trendu.

Nejprve bylo třeba vytvořit tabulku pomocí dotazu, který načte ID tiketu, datum vytvoření a datum uzavření, a také názvy firem a projektů pro účely filtrování.

Základem pro výpočet otevřených tiketů je opět datová tabulka. Jazyk DAX pro tyto účely poskytuje funkce *calendar()*, která vrací tabulku s jediným sloupcem obsahujícím data, jejichž rozsah je dán parametry ve funkci, nebo *calendarauto()*, která vrací stejnou tabulku, rozsah dat je ale vypočítán automaticky podle údajů v modelu. Vzhledem k množství dat, která se mění každý den je vhodnější zvolit druhou funkci. Datová tabulka tak bude vždy aktuální a bude obsahovat veškeré hodnoty, které se nacházejí v tabulce tiketů od nejstaršího data.

Dále už stačilo k datové tabulce správně přiřadit záznamy. Ty se přiřazují na stejném principu jako v prvním řešení – pokud se datum nachází mezi datem vytvoření a datem uzavření tiketu, je vložena 1. Tuto formuli bylo třeba napsat v jazyce DAX. Kód umí pracovat i s nulovou hodnotou, není tak třeba žádné dodatečné ošetření jako v případě prvního řešení.

---

```
Open Tickets = COUNTROWS(FILTER(ALL(Tabulka1); Tickets[Open].[Date] <= Dates[Date].[Date] &&
    Tickets[Closed].[Date] >= Dates[Date].[Date])))
```

---

Výpis 5: Přiřazení hodnot k datové tabulce

Tento způsob pro výpočet otevřených tiketů se mi zdál vhodnější vzhledem k automaticky aktualizované datové tabulce a z tohoto důvodu jsem jej využila pro vytvoření výsledného reportu.

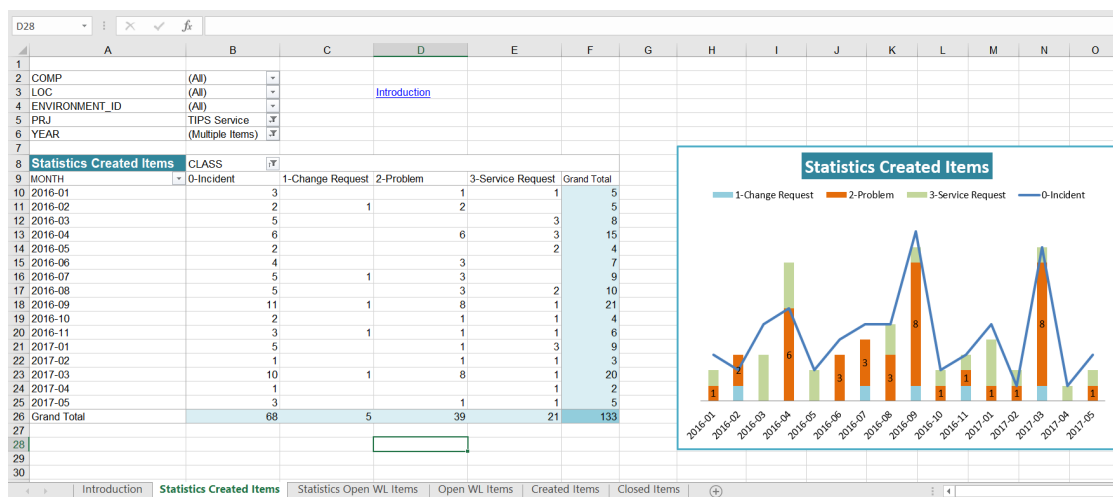
### 5.5.3 Výsledné řešení

Pro vytvoření výsledného reportu je potřeba jen spojit oba dotazy dohromady. Toho se dá dosáhnout v Power BI tak, že se v souboru vytvoří 2 tabulky, jedna pro výpočet vytvořených a uzavřených tiketů, druhá pro výpočet otevřených tiketů. Tyto dvě tabulky se spojí pomocí relací. Jelikož se relace dá vytvořit jedině pokud jedna z tabulek má jedinečné hodnoty, bylo třeba vytvořit další datovou tabulku, a také tabulky obsahující názvy firem a projektů. Na těchto tabulkách se poté vytvoří relace s tabulkami, které obsahují potřebné výpočty.

Jako filtry pro celý report se poté využívají sloupce z tabulek vytvořené pro relace. To umožní, aby jeden filtr ovlivňoval data ze všech tabulek. Pokud by nebyly tabulky propojeny pomocí relací, musel by filtr existovat pro každou zvlášť, což by bylo neefektivní.

## 5.6 Porovnání starého a nového řešení

Původní reporty byly získávány z excelovského souboru pomocí maker, které zajišťovaly jak přístup k databázi, tak vypsání tabulek a zobrazení grafů. Filtrování dat se provádělo na skryté kartě, na které bylo třeba vyplnit dvě data, mezi kterými se mají reporty zobrazit, a identifikační číslo firmy, pro jehož zjištění se uživatel musel přepnout do systému a najít odpovídající číselnou hodnotu. Reporty se vytvořily po stisknutí tlačítka, přičemž někdy nastala situace, kdy se prvky špatně rozmístily, což musel uživatel následně upravit ručně. Přístup k datům tak mohl zabrat až pět minut. Jednotlivé reporty poté byly rozmístěny na kartách. Pokud bylo třeba report poslat zákazníkovi, mohla nastat situace, kdy byl soubor příliš velký a přes e-mail se nepodařil odeslat.



Obrázek 5: Původní verze reportovacího systému

Výhodou původního řešení ovšem bylo jeho postavení na makrech, což umožnilo uživatelům vlastní úpravy reportu. Podmínkou však byly znalosti potřebné k jejich naprogramování. Nové řešení žádné takovéto změny nepodporuje.

Nové reporty jsou uživatelsky přívětivější, mají jednodušší ovládání. Přístup se provádí pomocí webového prohlížeče. Data se filtrují pomocí kliknutí, firma se vybírá z nabídky, není třeba znát identifikační číslo, navíc zde funguje vyhledávání dle názvu. Změna filtru ihned přepočítá veškerá data. Přístup k datům tak probíhá v rámci sekund.

Uživatelé chválí především interaktivní chování a rychlost přístupu k datům. Také je těší moderní vzhled. Pokud je zapotřebí odeslat report zákazníkovi, uživatel jednoduše nejprve data vyfiltruje podle potřeby a poté zákazníkovi odešle pouze screenshot reportu.

## 5.7 Budoucí funkce

Reporty se momentálně nacházejí na testovacím serveru, k němuž je možný přístup přes webovou adresu. Každý report je umístěn v samostatném souboru, jelikož se počítá s jejich implementací do programu Weblist. Implementace je ale bohužel prozatím v nedohlednu kvůli technickým problémům. Uživatelé se tak musejí přepínat mezi stránkami a nastavovat filtry pro každý report zvlášť.

Z tohoto důvodu momentálně pracuji na prozatímním řešení, ve kterém se veškeré reporty nacházejí v jednom souboru, kde je každá karta věnována jednomu reportu, podobně jako tomu bylo u původního řešení. Obrovskou výhodou pak budou sdílené filtry mezi všemi reporty. Efektivnější bude také rychlejší přepínání reportů pomocí karet. Uživateli pak odpadne nutnost několikrát nastavovat stejný filtr a vracet se vždy zpět na hlavní stránku pro zobrazení jiného reportu.

## 6 Další úkoly

Po dokončení reportů pro Weblist jsem dostala za úkol vytvořit widgety pro projekty ve VSTS. Tyto widgety byly vypracovány opět přes Power BI, jelikož jej VSTS bezproblémově podporuje a je přímo určen k průzkumu dat a vytváření vizuálů. Pro funkčnost je třeba mít nainstalováno rozšíření Analytics pro VSTS. Toto rozšíření poté umožní přístup k automaticky generované databázi, jejíž zjednodušená dokumentace se nachází na stránkách Microsoftu. Widgety byly vytvářeny pro projekty Tips Web a Capacity Planning.

### 6.1 KPI Widgety

Pro projekt Tips Web bylo třeba vytvořit widgety zobrazující KPI – klíčové ukazatele výkonnosti neboli klíčové metriky. KPI se používají k měření úspěšnosti aktivit, jejichž výsledné klíčové ukazatele by měly přímo ovlivňovat úspěšnost vize organizace.

U VSTS tyto ukazatele mají sloužit k informacím o úspěšnosti vývoje projektu. Vývoj tohoto projektu je postaven na metodě agilního vývoje softwaru – Scrumu. Práce, která je třeba vykonat se v takovémto případě rozdělí na části - sprinty, které mohou trvat např. týden, nebo měsíc. Jednotlivé úkoly v rámci sprintu se nazývají work items. Ty mohou být různého typu, například bug, product backlog item apod.

Pro načtení dat se dá využít datový konektor VSTS, který je k dispozici v Power BI prozatím v beta verzi. Takto získaná data jsou rozdělená jen podle typu položky, která je buďto work item nebo bug. Veškerá data jsou poté obsažena v jedné tabulce. Jelikož u tohoto typu konektoru nelze nastavit dataset a nelze tak žádným způsobem vybrat pouze data potřebná k zpracování, je načítání dat velice pomalé. Pro zrychlení a zefektivnění výběru dat jsem našla řešení, které využívá načtení dat z VSTS do Power BI pomocí OData. Při využití tohoto způsobu jsou data rozdělená na entity, mezi kterými existují relace. Entity obsahující data o work items jsou rozděleny podle potřeb pro různé druhy analýzy. Nejdůležitější jsou tabulky WorkItems<sup>4</sup>, WorkItemsRevision<sup>5</sup> a WorkItemSnapshot<sup>6</sup>.

Pro účely sestavení KPI ukazatelů bylo třeba využívat tabulku revizí, jelikož některé work items se mohou nacházet v několika různých sprintech najednou v jiném stavu, a pro účely správného výpočtu je třeba mít veškeré zpracovávané work items v daném sprintu. Kdyby byla využita tabulka WorkItems, mohly by být některé sprinty i úplně prázdné, protože se z daného sprintu posunuly do dalšího.

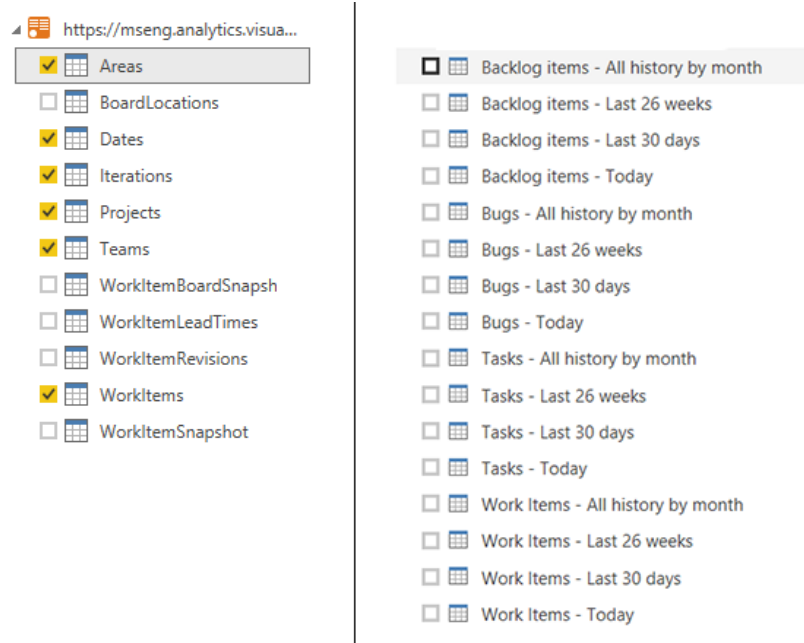
Pro zjednodušení jsem načetla pouze tabulku revizí, Power BI totiž umožňuje v návrhovém režimu do tabulky přidat další sloupce z jiných tabulek. Konkrétně bylo třeba přidat sloupce Iteration, Start date a End date z tabulky Iteration, které obsahují informace o názvu iterace(sprintu) a datum jejího zahájení a ukončení.

---

<sup>4</sup>Tabulka obsahující aktuální stav jednotlivých work items

<sup>5</sup>Tabulka obsahující veškeré historické revize work items, včetně aktuální

<sup>6</sup>Tabulka obsahující stav každého work itemu za každé kalendářní datum – určena pro trend analýzu



Obrázek 6: Porovnání tabulek při načítání pomocí konektoru OData (nalevo) a VSTS (napravo)

Dále bylo třeba pro přehlednost odstranit nepotřebné sloupce, což alespoň trochu dokázalo urychlit načítání dat. Spousta sloupců navíc obsahuje nulovou hodnotu.

Pro projekt Tips Web bylo třeba vytvořit KPI widgety počítající procentuální hodnotu úspěšnosti sprintu, přesnosti odhadovaných hodin potřebných pro splnění úkolu, neplánovaných chyb apod. Veškeré tyto výpočty se provádějí s work items, které jsou typu PBI. PBI znamená product backlog item, což je jednotka práce, která je dost malá na to, aby mohla být týmem dokončena během jednoho sprintu.

První a jedinný widget, který byl vytvořen skrze tento návrh řešení, je KPI, které slouží k vypočítání úspěšnosti sprintu. Jeho výsledná hodnota je dána tímto vzorcem:

*součet effort bodů<sup>7</sup> PBI, které jsou ve stavu done / součet effort bodů PBI, které jsou ve stavu committed*

Tento vzorec by se dal popsat jako součet všech effort bodů PBI dokončených na konci sprintu děleno součtem všech effort bodů PBI, které byly odhadnuty na začátku sprintu během meetingu. Zjednodušeně řečeno se jedná o poměr reálně vykonané práce ku původně naplánovanému množství práce.

Power BI obsahuje grafický prvek pro vytváření KPI grafů, ten je ale vhodnější použít za předpokladu, že se data třídí na základě nějaké množiny hodnot, např. data. Proto jsem se místo tohoto grafického prvku rozhodla použít číselnou kartu, která ukazuje pouze výslednou procentuální hodnotu pro zvolený sprint.

<sup>7</sup>práce, která je nutná udělat pro dokončení PBI

K výpočtu jsem využila funkci Power BI pojmenovanou míry, které slouží pro kalkulaci dat. Jelikož se tyto výpočty provádějí pomocí knihovny DAX, se kterou nedokážu příliš dobře pracovat, rozdělila jsem výpočet na tři míry, které představují dělence, dělitele a podíl.

---

```
Pocet za Done =  
CALCULATE(  
    SUM('WorkItemRevisions'[Pocet]);  
    'WorkItemRevisions'[State] IN { "Done" }  
)
```

---

Výpis 6: Kód pro součet effort bodů

Bohužel ani řešení, které načítá data pomocí OData protokolu vytváření widgetů v Power BI příliš nezeefektivnilo. Načítání bylo stále pomalé, informace v tabulkách jsou i v tomto případě velmi nepřehledné – položky mají například i dvě různé ID. OData ale neslouží pouze k tomuto účelu. Je vhodný i pro vývojáře, jelikož poskytuje přístup k veškerým datům, a díky tomu se dá využít i pro tvorbu vlastních widgetů nebo nástrojů.

Jelikož tvorba widgetů přes Power BI by trvala opravdu dlouho, navrhla jsem týmu řešení skrze programovací cestu. Takto si tým mohl zpracovat widgety sám podle představ.

## 6.2 VSTS reporty

Další widgety pro VSTS byly vytvářeny pro projekt Capacity Planning, kde bylo zapotřebí jiných typů reportů. Množství dat zapsaných pod tímto projektem není mnoho, a proto se daly widgety bez problémů zhotovit v Power BI, na rozdíl od komplexnějších výpočtů potřebných pro KPI.

Úkolem v tomto případě nebylo pouze zhotovení reportů, ale také najít vhodný způsob pro jejich publikaci přímo na dashboardu daného projektu.

Reporty jsou určeny k plánování sprintu. K tomu je třeba vizualizace, která slouží pro zobrazení množství hodin potřebných pro vývoj produktu oproti dostupnému volnému času (kapacity) vývojářů. Dále je třeba report pro retrospektivní porovnání naplánovaného a reálného času stráveném na vývoji.

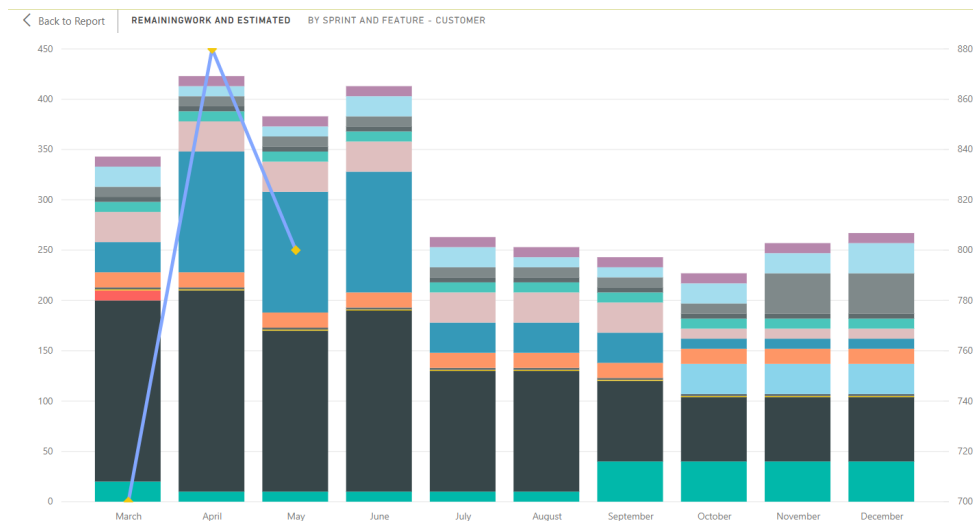
Pro přístup k datům byl opět vhodnější konektor OData vzhledem k jeho lepšímu rozdělení tabulek. Oproti KPI pro tyto reporty stačila pouze tabulka aktuálních stavů WorkItems, díky čemuž bylo načtení dat mnohem rychlejší a nepůsobilo takové problémy.

Pro vytvoření vizualizací nebyly třeba žádné výpočty, ale pouze načtení správných hodnot a jejich dosazení do grafu. Pracovala jsem hlavně se sloupci RemainingWork – naplánované hodiny, Logged – počet hodin reálně strávených na projektu a Estimated – dostupná kapacita vývojářů.

Jediná věc, kterou bylo potřeba dodělat, bylo správné seřazení měsíců (sprintů). Jelikož je sloupec s názvy sprintů typu text, umí jej Power BI srovnat pouze podle abecedy. Pro opravení tohoto problému jsem vytvořila tabulku s názvy měsíců a jejich číselným označením. Mezi tabul-

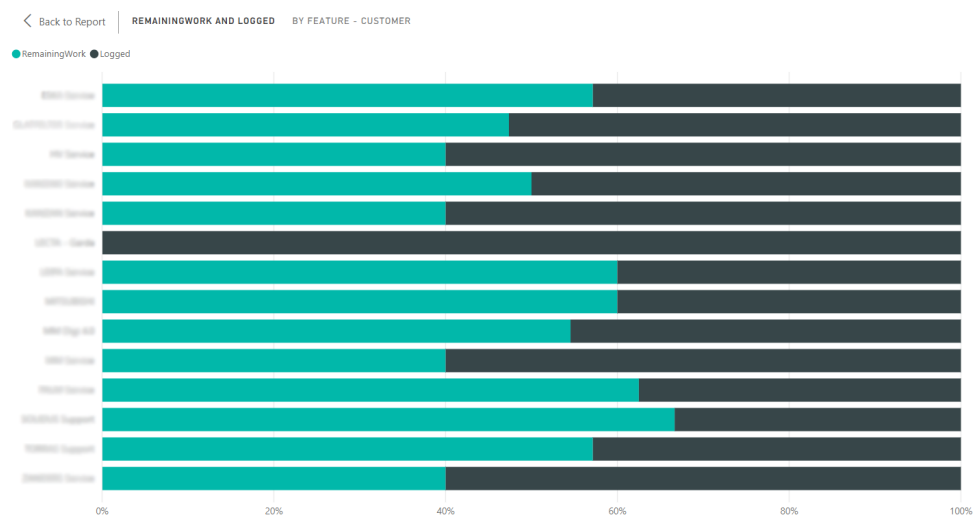
kami jsem poté vytvořila relaci a vytvořila tak další sloupec, který ke každému sprintu přiřadil pořadové číslo. Následně jsem sprinty seřadila podle tohoto nového sloupce.

První report zobrazuje kombinovaný sloupcový a spojnicový graf, kde sloupce znázorňují naplánované hodiny podle zákazníka za sprint a osa znázorňuje dostupnou kapacitu týmu. Data jsou navíc filtrována podle modulů.



Obrázek 7: Kombinovaný graf pro VSTS

Druhý report zobrazuje již zmíněné porovnání naplánovaných hodin oproti skutečnému času potřebnému pro vývoj podle zákazníka. Jinak řečeno jde o vizuální zobrazení hodnot ve sloupcích RemainingWork a Logged. Stejně jako předchozí report mohou být data filtrována podle modulů a navíc také podle sprintu.



Obrázek 8: Report porovnání hodin pro VSTS

## 7 Závěr

Absolvování individuální odborné praxe považuji za cennou zkušenost, jelikož mi to umožnilo náhled na práci v IT firmě. Díky této příležitosti jsem mohla uplatnit své vědomosti, které jsem získala v průběhu bakalářského studia. Především se jednalo o znalosti nabyté v databázově orientovaných předmětech, které jsem takto mohla ještě prohloubit.

Během praxe se mi podařilo vytvořit nový reportovací systém, prozatím přístupný pouze z webové stránky, který zrychluje předávání informací a má tak dopad na zvýšení efektivnosti práce v týmu. Systém je již využíván, uživatelé na něj přecházejí postupně, jelikož ne pro všechny jsou zatím informace dostačující a počítá se s budoucím rozšiřováním o další reporty podle potřeb uživatelů. Dále je také předpokládána jeho implementace do nástroje Weblist, aby tak byl přístup k datům ještě intuitivnější.

Také se mi podařilo najít a navrhnout vhodné řešení pro tvorbu podrobnějších widgetů pro dashboard VSTS, z nichž některé jsem vytvořila sama.

Praxe mi umožnila naučit se pracovat s novými technologiemi a prohloubila mou schopnost analytického myšlení. Naučila jsem se lépe pracovat na základě požadavků, a také najít to nejlepší řešení místo zvolení cesty nejmenšího odporu. Specialistů v oblasti business intelligence navíc není mnoho, o to více si této zkušenosti vážím.



## Literatura

- [1] Informace o Tieto. *Tieto - IT, výzkum a vývoj a poradenství*. [online]. Copyright © 2016 Tieto [cit. 24.04.2018]. Dostupné z: <https://www.tieto.cz/tieto-o-nas>
- [2] BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: podnik v informační společnosti*. 3., aktualiz. a dopl. vyd. Praha: Grada, 2012. Management v informační společnosti. ISBN 978-80-247-4307-3.
- [3] POUR, Jan, Miloš MARYŠKA a Ota NOVOTNÝ. *Business intelligence v podnikové praxi*. Praha: Professional Publishing, 2012. ISBN 978-80-7431-065-2
- [4] SQL Tutorial. *W3Schools Online Web Tutorials* [online]. Dostupné z: <https://www.w3schools.com/sql/default.asp>
- [5] Víte, co je SQL? Ne? Nevadí - dnes začínáme! – Živě.cz. *Živě.cz – O počítačích, IT a internetu* [online]. Dostupné z: <https://www.zive.cz/clanky/vite-co-je-sql-ne-nevadi—dnes-zaciname/sc-3-a-4320/default.aspx>
- [6] Reporting Services (SSRS). *Learn to Develop with Microsoft Developer Network / MSDN* [online]. Copyright © 2018 Microsoft [cit. 24.04.2018]. Dostupné z: [https://msdn.microsoft.com/en-us/library/ms159106\(v=sql.120\).aspx](https://msdn.microsoft.com/en-us/library/ms159106(v=sql.120).aspx)
- [7] Power BI | Výpočetní centrum Vysoké školy ekonomické v Praze. *Výpočetní centrum Vysoké školy ekonomické v Praze* [online]. Dostupné z: <https://vc.vse.cz/office365/power-bi/>
- [8] Data Analysis Expressions (DAX) Reference | Microsoft Docs. *Learn to Develop with Microsoft Developer Network / MSDN* [online]. Copyright © 2018 Microsoft [cit. 24.04.2018]. Dostupné z: <https://msdn.microsoft.com/en-us/query-bi/dax/data-analysis-expressions-dax-reference>
- [9] What is Open Data Protocol (OData)? Webopedia Definition. *Webopedia: Online Tech Dictionary for IT Professionals* [online]. Dostupné z: <https://www.webopedia.com/TERM/O/odata-open-data-protocol.html>
- [10] Describes the services provided by Visual Studio Team Services - VSTS / Microsoft Docs. [online]. Dostupné z: <https://docs.microsoft.com/en-us/vsts/user-guide/what-is-vsts?view=vsts>
- [11] TURLEY, Paul. *Professional Microsoft SQL Server 2012 Reporting Services*. Indianapolis, IN: John Wiley & Sons, 2012. ISBN 9781118262108
- [12] What is a KPI? Definition, Best-Practices, and Examples. *Business Dashboard Software for Everyone - Klipfolio* [online]. Dostupné z: <https://www.klipfolio.com/resources/articles/what-is-a-key-performance-indicator>